Wind River’s pLUG&SIM™ offers a flexible way in which two or more applications can participate in a cosimulation session, making pLUG&SIM a complete and open solution for any development team’s cosimulation needs. The simulation of hybrid models containing heterogeneous components such as software, hardware, hydraulics, electronics, and mechanical systems is often required. These models are typically developed using a variety of modeling and simulation tools from various vendors, making simulations of comprehensive systems dependent upon a variety of mechanisms.

pLUG&SIM is the first cosimulation framework to address traditional limitations such as one-to-one tool bridges, proprietary interfaces, and dependencies on tool versions, all of which have made cosimulation a difficult and time-consuming task. pLUG&SIM brings the designer a considerably more flexible and comprehensive way in which two or more applications can participate in a cosimulation. pLUG&SIM’s underlying communications architecture is based on CORBA, an open, vendor-neutral standard, ensuring that a cosimulation session can be executed on a single CPU or across a network on different platforms.

**Features**
- Graphical definition of simulation hierarchy based on model components
- Reuse of component definition tool simulators
- Extensible to any simulator through a standard CORBA API

**Benefits**
- Intuitive user interface
- Improved design process by bringing simulation to hybrid systems
- Supports any number of simulators
- Supports distributed computing
- Minimizes additional expertise requirements

**Supported hosts**
- Windows NT
- Solaris
Designing comprehensive systems
Systems are defined using components from many areas of expertise such as control system design, mechanical design, circuit design, or software design. Each of these component designs commonly requires a different simulation tool. Using these tools, development teams can independently design and simulate a specific component. pLUG&SIM then allows teams to integrate these component models into a comprehensive representation of a complete system.

A hybrid model definition requires a series of steps. Initially, the top-level model component in the Model Component View must be selected. For each of the undefined references in the top-level model component, another model component (possibly from another tool), must then be selected from the list of candidates in the hierarchy connection dialog. Consistency checking of the model’s input and output signals is performed at this point. If desired, the user can modify input and output signal routing, or force connections of unmatched signals. The process for model component selection may be repeated for any undefined references in the hierarchy. The hierarchy hybrid view allows the user to visualize the emerging hierarchy in an intuitive manner during its construction.

Simulating comprehensive systems
Once created, a hierarchical system model can be simulated in order to validate overall system design. The simulation and validation process requires the user to specify the simulation options for each tool participating in the cosimulation session. System behavior can be simulated and visualized using each component’s native tool. Finally, when a component’s native tool allows data to be saved, postsimulation data analysis can be performed in the native tool.

Simulation of large heterogeneous models is frequently a CPU-intensive task. pLUG&SIM, however, makes it possible to optimize simulation time by supporting simulation over networks, thus facilitating the use of fast hardware platforms.

Open architecture
pLUG&SIM is built on the industry-standard Common Object Request Broker Architecture (CORBA) object framework, which enables users to work in a distributed computing environment. The Cosimulation Using Distributed Objects (CUDOS) applications program interface (API) is a published API that enables third-party vendors to provide adapters for their particular simulators. pLUG&SIM does not place limitations on the simulators participating in a cosimulation session. This flexibility permits each simulation tool to participate with a different level of conformance.

At the parent conformance level, the tool can only participate as a top-level simulator. At the child conformance level, the tool can only participate as a bottom-level simulator. A tool can comply with both parent and child conformance levels, and tools can also support additional conformance levels for functions related to simulation rescheduling, including rollback, asynchronous events, and fixed/variable step integration algorithms. pLUG&SIM’s open architecture enables any tool vendor, or company with a legacy or proprietary simulator, to be integrated within the framework.
Improved design process
pLUG&SIM supports an iterative design approach that facilitates the early validation of a design. The hierarchical description of a system is always maintained. Systems can be simulated at any time during the design phase, either completely or partially. Moreover, simulators used for a stand-alone component design can also be used for a co-simulation session, ensuring continuity between component and complete system simulation.

Distributed resources
pLUG&SIM supports distributed computing. A simulation session can be performed on the same CPU or across a network on different platforms. Simulation can be optimized with respect to hardware performance, where CPU-intensive simulators can be placed on faster machines. In addition, pLUG&SIM facilitates the reuse of components within a project or between projects.

Minimum training
pLUG&SIM has an intuitive user interface for model hierarchy definition and simulation. Model descriptions are kept in their native tool. Analysis and post-analysis of simulation data is also performed within each component tool, limiting the need for training.

Simulation tool adapters
Several tools can work in conjunction with pLUG&SIM. For the latest available tool adapters, contact your Wind River sales representative.

pLUG&SIM allows easy access to model information. The emerging hierarchy can be visualized intuitively during model construction.